

SOKM/L语言在JSD知识表示 与利用中的应用

何克清 金明源 (武汉大学软件工程实验室)

摘要

JSD^[1] (Jackson System Development), method of structured software system design, was developed in 1980 by M. A. Jackson.

The basic views of JSD which identify with the ideas of Object-Oriented Development is that software system design ought to mirror the real world, i. e. the structure of software system would be coincided with that of problem domain of the real world.

This paper describes the procedural knowledge and the specification knowledge of JSD software development method with SOKM^[2] and SOKM/L^[3], which is well illustrated by an example.

Jackson 系统开发方法^[1] (Jackson System Development) JSD是1980年M.A. Jackson提出的结构化软件系统设计方法。

JSD的基本观点是软件系统的设计应该反映现实世界,也就是说软件系统的结构应该对应现实世界的问题领域结构。这与面向对象 (Object-Oriented) 的思想是一致的。

因此,本文利用SOKM^[2]和 SOKM/L^[3]描述JSD软件开发方法的过程性知识和说明性知识,并通过一个例子加以说明。

1. 基本概念

1.1 JSD的基本概念

实体 (entity), 一个实体是活跃的、动态的、通过所完成或所经历的动作及其时序表示的一个动作有序集。JSD中的实体必须按有意义的时序完成和经历动作。实体必须同时存在于现实世界的问题领域和软件系统内部。

动作 (action), 是在特定时刻瞬时发生而不是一段时间内延续的活动。与实体一样,动作必须同时存在于软件系统内部和现实世界的问题领域。实体的动作必须不可再分解为子动作。

实体和动作有密切联系,一个动作总是由一个或多个实体完成或经历。

模型与功能: 在JSD中,根据现实世界问题领域,首先建立相应的模型,以模型作为沟通问题领域和软件系统的桥梁,展开系统功能并确立系统范围。在此基础上,我们才进行系统功能分析与实现,如图1所示。

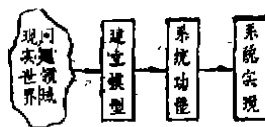


图1

系统模型对应于问题领域的结构,系统

功能基于系统模型,是模型中数据流等所对应的动作集。模型一经建立,是固定的,系统功能是可以增减的。

1.2 SOKM/L的描述

对象 (Object): 我们定义JSD中的实体为对象。对象活跃地、动态地完成或经历动作,是动作的集合。

动作: 对应于对象的方法 (Method)。

系统模型与功能: 我们定义JSD中系统模型为SOKM/L的对象模型,这是因为:

①实体、动作、对象对应现实世界的问题领域中个体,同时也是软件系统内部的单元;

②JSD系统模型与SOKM/L中系统模型都是沟通现实世界与功能系统的桥梁^[4]。功能是在系统模型基础上的展开,模型对于系统而言是相对固定的,功能是相对独立的,可增减的。

2. 开发规程知识的SOKM/L描述

2.1 JSD的开发规程

图2所示的树结构图表示JSD开发步骤,图中的树叶描述了相应实体的动作。

JSD规程的基本想法是从问题领域出发,逐步通过一些进程阶段建立一个可执行的模拟现实世界的计算机软件系统。即,

视现实世界问题领域为No.0进程;

系统模型为No.01进程;

系统功能为No.02进程;

.....

直至实现上述目标。

为了保证后进程正确反映前进程,我们遵循如下二条原则:

a. 结构对应: 模型中实体的结构与本节②中所抽出的实体结构相对应;

b. 事件对应: 为保证模拟的行为能正确反映现实世界中的事件,我们以数据流、状态向量方式联系各进程。

在建立上述各进程以及进程间的转换过程中,知识库和人工干预技术起了关键性的

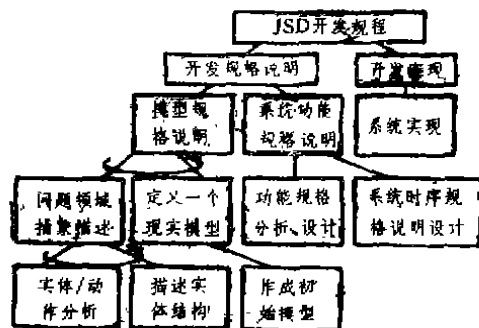


图3 JSD规程

作用。

下面我们对图2中六个叶子部分进行详细讨论。

①动作与实体的分析、抽出 根据实体与动作的定义,分析并抽出实体、动作,列出实体表与动作表,然后创建和产生系统中所有的类对象以及这些对象中应有的方法(method)。当然此步还不能产生出这些方法的具体内部结构。

我们以电梯控制为例^[5]加以说明:

【例】一栋六层楼安装电梯,除最高层外每一层有一个请求向上的电钮(U_1-U_5),除底层外每一层有一个请求向下的电钮(D_1-D_5),电梯内部有六个标识层号的电钮(E_1-E_6)。试分析该规格说明。

本例中实体与动作分析、抽象如下:

实体: ELEVATOR, BUTTON

动作: ELEVATOR的动作:

ARRIVE(m) 到达第m层

LEAVE(m) 离开第m层

BUTTON的动作:

PRESS 按下、放开一次

②实体结构 以结构树图方式定义各实体的内部结构。在此基础上产生各对象内部的slot结构,attribute结构,这就相当于构造了实体的结构(包括动作间的时序)。

本例中实体结构如图3所示:

③系统初始模型 本步在前2步的基础上,产生各对象内部的继承结构。然后以JSP的设计思想,根据问题领域的层次结构,使用顺序、重复和循环这三种结构形式初步

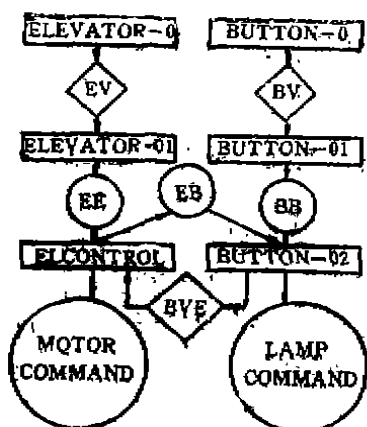


图6 系统进程关联

某层,然后发出电机命令,为此设一数据流EE。ELEVATOR-01向此数据流写ARRIVE(m)记录,ELCONTROL读取该数据流。

ELCONTROL在到达某一层时,要向BUTTON-01传递信息以便使BUTTON-01发控制键灯的命令。但现在的BUTTON-01不能适应这种功能,所以我们再定义一个二级进程BUTTON-02,通过数据流BB和BUTTON-01联系,如图6所示。

至此,我们完成了系统模型、功能进程及其相互作用关系的分析与设计,并得到了它们的图式表示与正文的规格说明。JSD的系统分析与设计工作基本完成。

⑤系统时序规格设计 在已完成的系统整体SSD图基础上设计与产生系统的约束对象^[2]和对象内部的约束结构,从而完善整个系统。系统的时序约束对象有:

- 输入;
- 状态向量的操作;
- 进程间同步的时序。

⑥系统实现 本步的主要工作是系统初始化和系统初启。例如就电梯系统来说,经过前五步,整个电梯系统的静态文本已建立起来了,但还不能运行。经过此步给系统初始化,然后将控制权移交给电梯对象中的操作约束而实行系统初启。这时整个系统不再是孤立的ELEVATOR和BUTTON等对象静态

地躺在那里,而是一个活生生的动态的电梯模拟系统在那儿运行。

2.2 SOKM/L的描述

SOKM/L语言能较容易地表示出图2中给出的JSD的开发规程和步骤,其表示如下:

JSD开发规程(Spec-in, Prog-out):-

{开发规格说明 (Spec-in, SSD),
开发实现(Spec-in, SSD, Prog-out)}.

开发规格说明(Spec-in, SSD):-

模型规格说明 (Spec-in, SSD1),
系统功能规格说明 (Spec-in, SSD1, SSD).

模型规格说明 (Spec-in, SSD1):-

{问题领域抽象描述(Spec-in, Jspd),
定义一个现实模型(Spec-in, Jspd, SSD1)}.

问题领域抽象描述 (Spec-in, Jspd):-

{实体-动作分析(Spec-in, Entity, Action),
概括实体结构 (Spec-in, Entity, Action, Jspd)}.

实体-动作分析(Spec-in, Entity, Action):-

{对象描述-分析(Spec-in, Entity),
动作描述-分析 (Spec-in, Entity, Action)}.

对象描述-分析 (Spec-in, Entity):-

{名词抽出 (Spec-in, Entity1),
{对象选择 (Spec-in, Entity1, Entity2),
对象综合-改善 (Spec-in, Entity2, Entity)}}.

对象综合-改善 (Spec-in, Entity2, Entity):-

“根据Spec-in和Entity2, 创建系统中所有有关类对象于Entity中。”

动作描述-分析(Spec-in, Entity, Action):-

{动作抽出 (Spec-in, Action1),
{动作选择 (Spec-in, Action1, Action2),
动作综合-改善 (Spec-in, Entity, Action2, Action)}}.

动作综合-改善 (Spec-in, Entity, Action2, Action):-

“根据Spec-in、Entity和Action2定义各对象应有的method于Action中。”

描述实体结构 (Spec-in, Entity, Action, Jspd):-

“根据Spec-in, Entity和Action产生各对象的slot结构和attribute结构于Jspd中。”

定义一个现实模型 (Spec-in, Jspd, SSD1):-

作成初始模型 (Spec-in, Jspd, SSD1).

```

作成初始模型(Spec-in, Jspd, SSD1):-
inherited(Spec-in, Jspd, Jspd1),
{顺序结构(Spec-in, Jspd1, SSD1)◇
重复结构(Spec-in, Jspd1, SSD1)◇
循环结构(Spec-in, Jspd1, SSD1)}.
inherited(Spec-in, Jspd, Jspd1):-
    "产生各对象的inheritance结构。"
系统功能规格说明(Spec-in, SSD1, SSD):-
    功能规格分析-设计(Spec-in, SSD1, SSD
    2),
    系统时序规格说明(Spec-in, SSD2, SSD).
功能规格分析-设计(Spec-in, SSD1, SSD):-
    {嵌入型功能设计(Spec-in, SSD1, SSD11)
    ◇
    强制型功能设计(Spec-in, SSD1, SSD11)◇
    相互作用型功能设计(Spec-in, SSD1, SSD-
    11)}
    构造method的具体结构(Spec-in, SSD11,
    SSD12),
    建立对象间的消息通讯结构(Spec-in, SSD12,
    SSD).
嵌入型功能设计(Spec-in, SSD1, SSD11):-
    {抽出操作(Spec-in, SSD1, Operator),
    嵌入操作(Spec-in, SSD1, Operator, SSD-
    11)}.
强制型功能设计(Spec-in, SSD1, SSD11):-
    定义功能进程(Spec-in, SSD1, SSD11).
相互作用型功能设计(Spec-in, SSD1, SSD):-
    分析模型与功能进程的相互作用(Spec-in,
    SSD1, Command),
    修改-完善SSD(Spec-in, SSD1, Command-
    SSD11).
构造method的具体结构(Spec-in, SSD11,
    SSD12):-
    {建立并构造local method(Spec-in, SSD11,
    SSD12)◇
    回溯技术(Spec-in, SSD11, SSD12)◇
    结构冲突消除技术(Spec-in, SSD11,
    SSD12)◇
    程序变换技术(Spec-in, SSD11, SSD12)◇
    :
    }.
回溯技术(Spec-in, SSD11, SSD12):-
    选择部分(Spec-in, SSD11, SSD111),

```

```

收容部分(Spec-in, SSD111, SSD112),
副作用处理(Spec-in, SSD112, SSD12).
副作用处理(Spec-in, SSD112, SSD12):-
    {局部变量法(Spec-in, SSD112, SSD12)◇
    变量转存法(Spec-in, SSD112, SSD12)◇
    设置恢复点法(Spec-in, SSD112, SSD12)}.
结构冲突消除技术(Spec-in, SSD11, SSD12):-
    {中间文卷法(Spec-in, SSD11, SSD12)◇
    多进程法(Spec-in, SSD11, SSD12)◇
    :
    }.
系统时序规格说明(Spec-in, SSD2, SSD):-
    {输入时序设计(Spec-in, SSD2, SSD21)◇
    状态向量的操作时序设计(Spec-in, SSD2,
    SSD21)◇
    进程间同步的时序设计(Spec-in, SSD2,
    SSD21)}
    创建系统的约束对象(Spec-in, SSD21
    SSD22),
    建立各对象内部约束结构(Spec-in, SSD22,
    SSD23),
    完善系统(Spec-in, SSD23, SSD).
开发实现(Spec-in, SSD, Prog-out):-
    作成实现模型(Spec-in, SSD, Sid),
    系统实现(Spec-in, Sid, Prog-out).
系统实现(Spec-in, Sid, Prog-out):-
    系统初始化(Spec-in, Sid, Prog1),
    系统初启(Spec-in, Sid, Prog1, Prog-out).
系统初启(Spec-in, Sid, Prog1, Prog-out):-
    "将控制权移交给系统中关键的约束对象."
    /* Spec, 自然语言书写的规格说明书
    SSD, System Specification Diagram
    Sid, System Implementation Diagram
    Jspd, Jsp diagram
    */

```

为了表示JSD开发规程的整个过程，我们定义六个步骤的略称为：

- ①ent/act, entity/action step
- ②struct, entity structure step
- ③model, Initial model step
- ④function, function step
- ⑤time, system timing step
- ⑥impl, implementation step

于是有定义:

$JSD ::= \{ \{ \{ \text{ent/act} \} \text{ struct} \} \text{ model} \}$

function time impl

JSD高层活动的frame表示请参见[2]。

①对象的分析与抽出

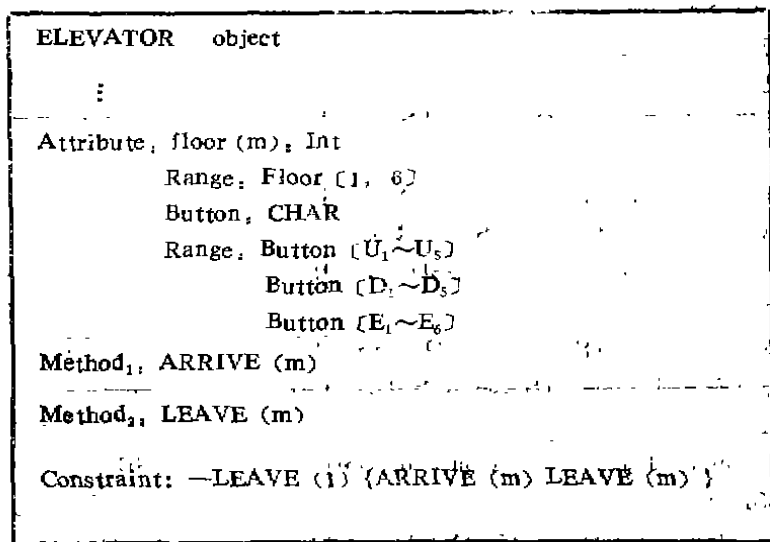
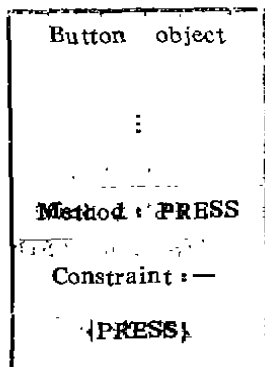


图7 对象的结构

③初始模型

图4给出了例题“电梯控制”的初始模型,在SO KM/L描述中,图8为例题的初始对象模型。其中,对象间消息传递对应着数据流、状态向量的传递。图8中采用SOKM/F[6]描述。

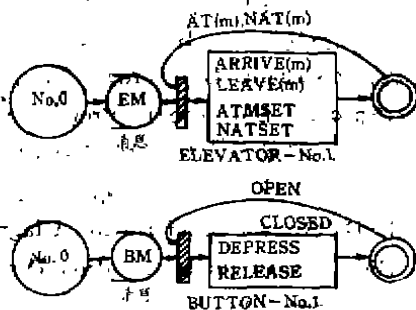


图8 初始模型

④功能规格分析、设计

根据2.1节④中所述, SOKM/L表示如下:

嵌入型功能:分析功能所对应的操作,对每一操作,采用SOKM/L中的方法(Method)、谓词、子句进行描述,然后嵌入到如图8所示的初始模型中去。

强制型功能:分析产生功能的条件,如属于外生事件(例如即席查询),则定义新的功能进程(对

与2.1节①同样,我们抽出对象ELEVATOR和BUTTON对象。

②对象结构

根据图3,对象ELEVATOR和BUTTON构造如图7所示:

象、消息),即新的类(class)。

相互作用型功能:分析模型进程与功能进程之间的相互关联的数据流状态向量,定义消息及消息传递。从而各进程及其间的消息传递构成系统功能规格说明。

功能进程所对应的对象(Object)可以看作是模型进程所对应的对象(Class)的实例对象(instance)。

对于上述三类功能的判断,都依赖于产生功能的条件及其判别。为此,必须设立三类功能条件判别的规则。

例题“电梯控制”中,相互作用型功能分析,设计如图9所示。其中消息内容与图8相同。

消息定义模式如下:

MSG(object name, slotname, 参数), 其中,对象名为ELEVATOR类名,BUTTON类名,以及ELEVATOR-01名,EICONTROL名,或BUTTON-01,BUTTON-02名,slot名为操作名,如ARRIVE,LEAVE等;参数为m,数据流、状态向量等。

至此,系统的需求定义与分析,功能设计工作告东结束。

⑤系统时序规格设计

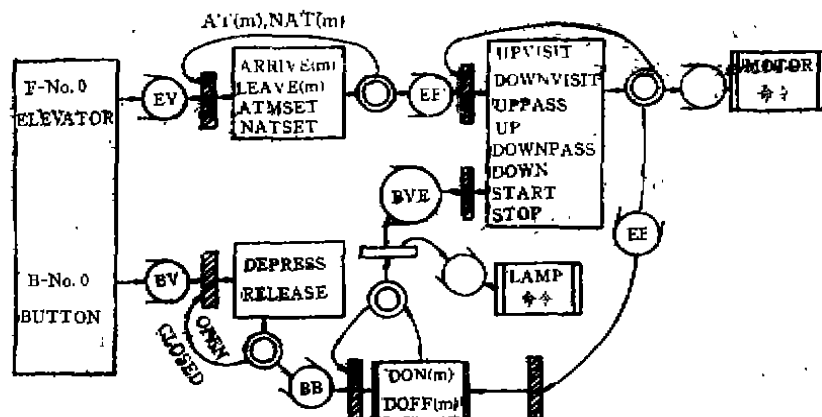


图9 系统功能规格图

对上述设计结果,运用约束手段对输入、状态向量操作、对象间消息传递的同步时序进行设计。

⑥系统实现

确定初始对象与消息流,实现对应系统。

作为附录,我们给出例题“电梯控制”的SOKM/L的正文描述。

3. 进一步的研究工作

本文利用SOKM/L概要地描述了JSD的开发规程和步骤,但是,并没有给出详细的规则的描述。为了比较充分地表示和利用JSD知识,必须进一步研究如下一些问题。

a. JSD开发规程(图2)中每步(图中每框)的规则细化描述。

b. 步骤(图中框)间转化规则的表示。例如,判断“对象分析、抽出”的合理性的规则;判断初始模型合理性的规则(也就是JSD转入系统功能分析、设计步骤的条件),也就是步骤间转化的GOTO规则研究与表示。

c. JSD知识系统的内部谓词的研究;

d. JSD知识系统的专门谓词的研究;

e. JSD知识系统中消息模式的研究;

f. JSD方法的软件开发环境的研究。

附录:例题“电梯控制”正文

ELEVATOR class

nature /* 没有上属继承类*/

attribute

m: int

Button D: int

Button U: int

Button E: int

Constraint

V-C: $m(1 \leq m, m \leq 6)$;

V-C: Button D ($1 \leq \text{Button D}, \text{Button D} \leq 6$); ($\text{Button D} \neq 0$);

V-C: Button U ($1 \leq \text{Button U}, \text{Button U} \leq 6$); ($\text{Button U} \neq 0$);

V-C: Button E ($1 \leq \text{Button E}, \text{Button E} \leq 6$); ($\text{Button E} \neq 0$);

Method

:ARRIVE(m):—ButtonE=0, WAIT;
/*电梯正在进行上下动作*/

:ARRIVE(m):—ButtonE=m;

/*此时电梯已在m层楼,不作什么*/

:ARRIVE(m):—ButtonE<m, ARRIVE(m-1), LEAVE U(m-1), STOP,

ButtonE<m;

:ARRIVE(m):—ARRIVE(m+1), LEAVE D(m+1), STOP, ButtonE<m;

:LEAVE D(1):—ERROR; /*出错处理*/

:LEAVE D(m):—ButtonE<m;

:LEAVE D(m):—ButtonD<m, Down, START;

:LEAVE U(6):—ERROR/*出错处理*/

:LEAVE U(m):—ButtonE<m;

:LEAVE U(m):—1, ButtonU<m, UP, START;

Constraint

M-C: LEAVE(1), {ARRIVE(m), LEAVE U(m), LEAVE D(m)}

local

```

UP:—ButtonE<=0, ButtonD<=D,
Down:—ButtonE<=0, Buttonu<=0;
START:—Button:PRESS(open);
/*该子句中用了一个类方法调用PRESS
(open)*/
STOP:—Button:PRESS(close);
/*该子句中用了一个类方法调用PRESS
(close)*/
WAIT:—...,
ERROR:—...,
end /* ELEVATOR */
Button class
nature
    APO:ELEVATOR †
attribute
method
    :PRESS(open):—RELEASE;
    :PRESS(close):—DEPRESS;
constraint†
    M-C:{PRESS(open), PRESS(close)}
local
    RELEASE:—..., DON(m):—...,
    DEPRESS:—..., DOFF(m):—...,
end /* BUTTON */

```

参考文献

- [1] M. A. Jackson, J. R. Cameron JSP and JSD: The Jackson approach to Software Development. 1983. IEEE
- [2] 何^[1]软件开发知识的表示与利用模型SO-KM 武汉大学“863”高新技术研究报告 No.05 1988
- [3] 何、金^[2] SOKM/L的设计 武汉大学 “863”高新技术研究报告No.08 1989.
- [5] 何、吕^[3] WUDA规范——面向对象的软件需求分析与定义方法 武大学报(自然科学版) CSE专利 1988.6.
- [5] 胡^[4] JSD——一种系统开发方法 计算机工程与应用 1986.1
- [6] 何^[5] 基于SOKM的图式语言SOKM/F 武大“863”高新技术研究报告No.07. 1988^{*} 12
- [7] 服部隆^[6] “SIMPOSスーパーバイザの対象指向型アプローチによる実現(オブジェクト指向); 日本共立出版(株) 1985.

第九届全国数据库学术会议征文通知

经中国计算机学会软件专业委员会同意,由中国数据库专业学组委托上海第二工业大学、公安部第三研究所(上海)、复旦大学、华东计算技术研究所举办第九届全国数据库学术会议。会议定于1990年9月10日至14日在上海召开。

内容:1.数据模型与语言;2.数据结构与存储方法;3.数据库设计方法与应用;4.数据库工程与应用;5.数据库理论与算法;6.用户接口;7.分布式数据库;8.知识库系统;9.多介质数据库;10.面向对象数据库/知识库。

日期:征文截稿:1990.3.30;录取通知:1990.4.25

稿件投寄:200041 上海陕西北路80号上海第二工业大学计算机系第九届全国数据库会议会务组朱宁收。

中国计算机学会数据库学组

第三届全国知识工程研讨会

由中国计算机学会、软件专业委员会、智能软件学组主持,四川大学、四川省计算机学会和成都科技大学等单位承办的第三届全国知识工程研讨会将于1990年9月上旬在四川省松潘召开。

内容:以知识为基础的系统;知识工程中的软件工具;知识获取与机器学习;推理;自然语言分析模型和分析方法;人工智能语言;并行及分布式知识处理;认知模型;基础研究。

中国计算机学会、软件专业委员会、智能软件学组